Efficient Multiclass Object Detection: Detecting Pedestrians and Bicyclists in a Truck's Blind Spot Camera

Kristof Van Beeck and Toon Goedemé
EAVISE, Technology Campus De Nayer, KU Leuven, Belgium
{kristof.vanbeeck, toon.goedeme}@kuleuven.be

Abstract

In this paper we propose an efficient detection and tracking framework targeting vulnerable road users in the blind spot camera images of a truck. Existing non-vision based safety solutions are not able to handle this problem completely. Therefore we aim to develop an active safety system, based solely on the vision input of the blind spot camera. This is far from trivial: vulnerable road users are a diverse class and consist of a wide variety of poses and appearances. Evidently we need to achieve excellent accuracy results and furthermore we need to cope with the large lens distortion and extreme viewpoints induced by the blind spot camera. In this work we present a multiclass detection methodology which enables the efficient detection of both pedestrians and bicyclists in these challenging images. To achieve this we propose the integration of a warping window approach with multiple object detectors which we intelligently combine in a probabilistic manner. To validate our framework we recorded several simulated dangerous blind spot scenarios with a genuine blind spot camera mounted on a real truck. We show that our approach achieves excellent accuracy on these challenging datasets.

1. Introduction

Each year traffic accidents caused by the blind spot zone of trucks are responsible for an estimate of about 1300 casualties in Europe alone. Since only accidents involving victims are reported, this figure is a great underestimation of the real problem. Several commercial systems have been developed that try to cope with this problem, ranging from simple mechanical solutions (*e.g.* blind spot mirrors) to more advanced automatic alarm systems. However, none of these systems seem able to adequately decrease the number of victims. Indeed, research indicates that the number of casualties did not decrease since the use of blind spot mirrors was obliged by law in 2003 in Europe [13]. This is mainly due to two reasons: most of these mirrors are not adjusted correctly and rely on the attentiveness of the driver. The first problem is solved using a robustly mounted blind spot camera, and a monitor in the truck's cabin. Evidently, the success rate of such a system again highly depends on the alertness of the truck driver. These systems are coined *passive*, whereas *active* systems automatically generate an alarm, such as ultrasonic distance sensors. The main disadvantage of the latter are *false positive* alarms; they are unable to distinguish vulnerable road users from static objects (*e.g.* traffic signs). The truck driver experiences this as annoying and therefore avoids the use of this system altogether. In this paper we propose a detection framework that overcomes the aforementioned problems: we developed a vulnerable road user (VRU) detection system based solely on the monocular blind spot camera images. Such a system has multiple advantages: it is always adjusted correctly, requires no truck driver interpretation and is easily implementable in existing passive blind spot camera solutions. Developing such a complete system however is challenging since vulnerable road users are a diverse class (pedestrians, bicyclists, mopeds, wheelchair users and so on), all with varying appearances. Our framework tackles the multiclass detection of both pedestrians and bicyclists, which are involved most in these type of accidents. Aside from this, the typical commercial blind spot cameras employ wide-angle lenses and thus introduce non-standard viewpoints and severe lens distortion which make it unfeasible to utilise out-of-the-box object detection algorithms. Due to the sideways-looking view a highly dynamical background is observed making it hard – or even impossible – to perform an initial segmenta-

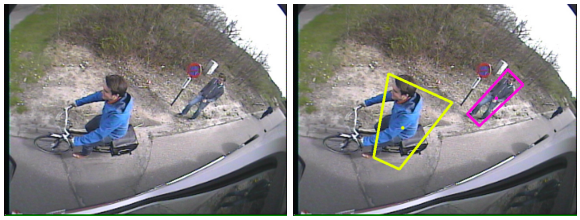

Figure 1. (Left) Example frame from our blind spot camera. Both high lens distortion and a non-standard viewpoint are observed. (Right) Output detections of our framework.

tion. Furthermore this application inherently requires a high detection accuracy. Fig. 1 displays an example frame of our dataset (left) – which indicates the complexity of these images – and the output of our framework (right).

The main contributions in this paper are two-fold. We give an approach to efficiently combine multiple object detectors using a probabilistic manner for these non-trivial images, and we propose a methodology which selects the most appropriate model to evaluate based on the position in the image. In a nutshell, our framework works as follows. First we employ a *warping window* approach: at each position in the input image we locally model the transformation due to the viewpoint distortion. Using this information we can rewarp each region of interest, effectively undoing the local distortion. Next we extract image features on this rewarped patch and generate probability maps for multiple object models, selected again based on the position in the image. These hypothesis maps are then combined into a single detection probability map for that image patch. Finally, to cope with missing detections we integrate these detection maps in a tracking-by-detection methodology.

To validate our approach we recorded several simulated dangerous blind spot scenarios with a genuine blind spot camera mounted on a real truck. These datasets involve both pedestrians and bicyclists - see section 4 for more information. Our algorithm achieves excellent accuracy results on these challenging datasets. The remainder of this paper is structured as follows. In section 2 we discuss existing work on this topic. Next we describe our algorithmic approach in detail in section 3, and provide both qualitative and quantitative evaluation results in section 4. Finally, we conclude this paper in section 5.

2. Related Work

A vast amount of literature on pedestrian detection is available, see [2] for a recent extensive overview. In essence, two popular approaches exist: *deformable part-based* models (DPM) and *rigid* models. Both methodologies are inspired by [5] where the authors presented the use of *Histograms of Oriented Gradients* (HOG) for pedestrian detection. These part-based models, introduced in [11], extended the rigid HOG model with *parts* representing *e.g.* the limbs and head of a pedestrian. Specific deformations for these parts are allowed - subject to a deformation cost - resulting in an increased detection accuracy. These DPM models remained among the top performing methods for several years [9]. Aside from the inclusion of multiple parts to increase the detection accuracy, [8] presented an approach that enriches the rigid model with additional features, coined the *Channel Features* detector. Multiple optimisations have been proposed to speed-up detection and increase the accuracy [1, 7]. The *Aggregated Channel Features* (ACF) detector [6] currently is one of the top performing detectors [2]. Recently, *deep learning* methods have become increasingly popular as a mains to further increase the detection accuracy. Indeed, using *convolutional neural networks* (R-CNN) unprecedented accuracy results are obtained [12]. This technique is able to simultaneously classify a large variety of classes, making it ideal for large image database retrieval applications such as ImageNet [14]. However, these methods rely on large databases for training and extensive hardware resources, rendering them currently unfeasible for real-time applications. Several works exist which apply the aforementioned algorithms on traffic safety applications, and are thus related to our work. However, to the best of our knowledge, often only forward-looking cameras are used and only single object classes (pedestrians or bicyclists) are detected [3, 4, 16]. In [15] we presented a similar safety system, however only targeting pedestrians which evidently is a much easier scenario. We differ significantly from all of these works: we aim to develop an efficient framework that enables the detection of multiclass objects in camera images with non-standard viewpoint and high lens distortion (see fig. 1).

3. Algorithmic Approach

Traditional object detectors employ a *sliding window paradigm*: a full scale-space feature pyramid is constructed and evaluated at each location. Such an approach is infeasible for our application. As seen in the example frame (fig. 1), the vulnerable road users appear under various rotations and scales. If we ought to run a standard object (*e.g.* pedestrian) detector on these images they need to be evaluated at multiple locations, scales and orientations which is impossible to compute in real-time. Moreover, due to the viewpoint and high lens distortion the detection accuracy will be suboptimal. To cope with these challenges we first employ a warping window approach, similar to our previous work [15]. We exploit the fact that the exact transformation (that is, rotation, scale and perspective effects) only depends on the position in the image. At each position in the image we thus locally model this transformation, and rewarp the *regions of interest* (ROI) to an undistorted, upright and fixed scale image patch avoiding the need to compute a full scale-space pyramid. Since we aim to detect both pedestrians and bicyclists, in the next step we extract features and run multiple detection models on these image patches. To reduce the computational complexity we employ feature sharing and only run specific models at specific locations. These detection maps are then combined into a single probability map. Finally we integrate this information in a tracking-by-detection framework. Figure 2 gives an overview of our detection approach. Note that due to space constraints the processing steps are shown for a single ROI only, in practice each ROI is validated. Let us now discuss each of the consecutive steps of our detection pipeline in detail.

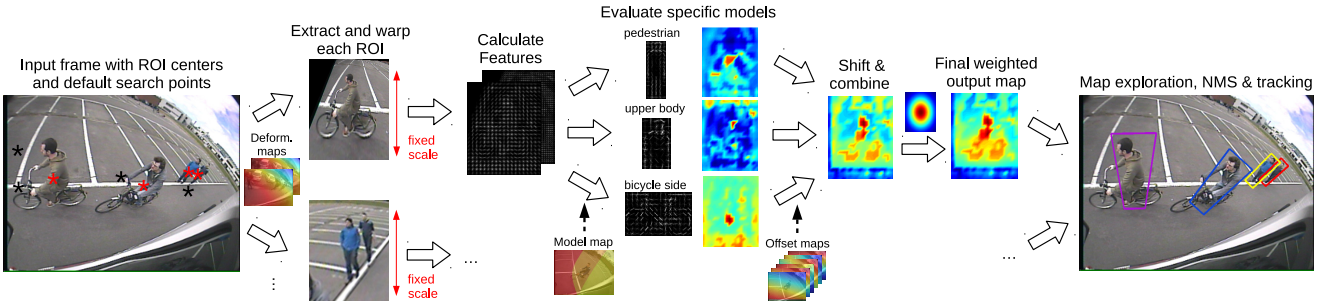

Figure 2. Overview of our algorithmic approach. Note that the probability maps displayed here are interpolated for visual purposes; during computation they are calculated discretely on a grid with a step size of 8 pixels (best viewed in color).

3.1. Warping patches

As previously mentioned the pedestrians and bicyclists appear rotated, scaled and distorted at specific positions in the image. This transformation only depends on the position in the image, if we assume a flat groundplane – which evidently is a valid constraint in our application. Thus, if this transformation is known, each ROI can be rewarped to an upright position at a fixed scale. This approach evidently eliminates the construction of a full scale-space pyramid, and allows the use of a single upright detection model (for each class). Since only evaluation at a single scale is performed, this approach allows the use of an accurate detector which would otherwise be too time-consuming to run in real-time. A different approach is possible where we directly train the detection models using the distorted images. However, in this case a vast amount of new training data is needed when a different blind spot camera is used. Using our approach only a basic recalibration is required.We modelled this distortion as a perspective transformation. The local deformation for each position is extracted in an offline step, and stored in two *deformation maps* as visualised in the left of fig. 2 (see [15] for details). We thus model the pedestrians as planar objects, faced towards the camera. Our experiments indicate that this is a valid assumption for pedestrians. For bicyclists, this assumption is not valid at all positions in the image. However, this concern is tackled further in our detection pipeline: we evaluate multiple bicycle viewpoint models depending on the position in the image. During detection we employ these deformation maps and the vantage line to effectively undo the local rotation and perspective transformation, and warp the ROIs to a fixed scale of 140 pixels, as this has proven to be an adequate trade-off between accuracy and computational complexity. These upright, fixed-scale image patches are then fed in to our detection pipeline.

3.2. Object detection pipeline

The unwarped image patches can now be processed to detect both pedestrians and bicyclists. Several object detectors were discussed in section 2. Currently, rigid detectors are slightly more accurate as compared to deformable part-based model approaches [2]. However, the advantage of the latter is of strong importance in our framework: since deformation is allowed, slight deviations from the trained model and the object to be detected are tolerated. Since we only perform detection at a single scale this deformation is essential: multiple scales are needed with a rigid model to achieve accurate detection results. Therefore we opted to use the cascaded DPM [10] as a baseline. In a first step, for each ROI image patch we extract a 31 dimensional feature vector (consisting of HOG and contrast features). Since the detection accuracy increases if the features for the different parts are calculated more densely [11], this is done for two bin sizes. To robustly detect both pedestrians and bicyclists we share these features between different detection models. At each position in the image we validate three models: a pedestrian model (trained on INRIA), an upper body model and a bicycle model (both trained on the VOC dataset). Evidently, we trained all models such that the size of their root models are equal, and chose to utilise 8 parts. The pedestrian and upper body model consist of a single component (i.e. a single viewpoint). However, the bicycle model consists of three components: a frontal, semi-side and sideways looking viewpoint. Based on the position in the image we perform *model selection*: we only select the single, most optimal bicycle detection component to run at that location and thus decrease the calculation time. For this, in an offline phase we evaluated all three components on labelled bicy-

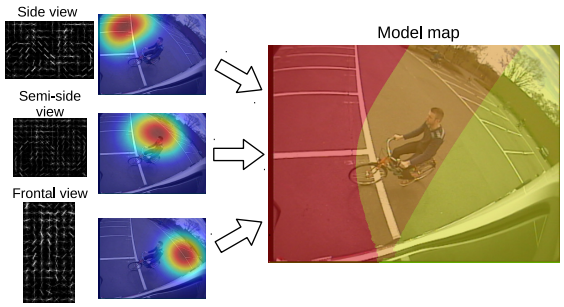

Figure 3. Generation of the *Model map* which indicates which bicycle component should be evaluated where in the image.

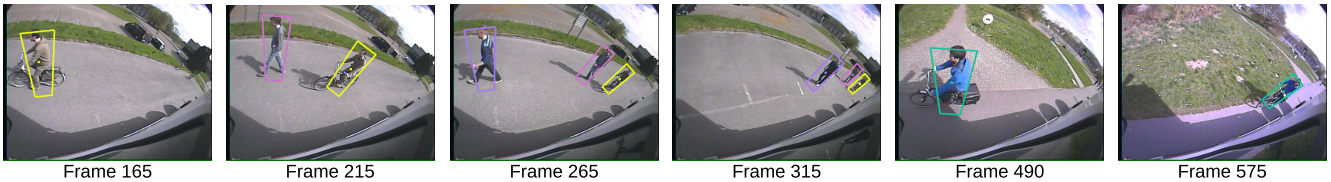

Figure 4. A qualitative tracking sequence over one of our datasets. See http://youtu.be/0xFdDOYxKK8 for a video.

clists homogeneously spread over the image and selected the best scoring for each image position. With this data we generated a probability map for each component and combined these maps into a final image segmentation, as shown in fig. 3. This final *Model map* thus indicates which bicycle component should be evaluated at each position. These three models (pedestrian, upper body and one of the bicycle components) - and their mirrored versions (we take the maximum of both) - are evaluated on a grid of 8 pixels, and yield three discrete probability maps for each image patch.

3.3. Combining probability maps

Finally, these probability maps $P_i(\mathbf{x})$ with $\mathbf{x} = [x, y]$ for each component i are combined into a single probability map using:

$$P_{final}(\mathbf{x}) = \max_{\substack{\mathbf{x} \\ i \in \{1,2,3\}}} (P_i(\mathbf{x}) - d_i(\mathbf{x})) + G(\mathbf{x}) \qquad (1)$$

Here, $d_i(\mathbf{x})$ indicates an offset for each component used to ensure correct detection localization. For this we shift each map such that the expected maximum of the detection models coincides which each other (*e.g.* the upper body model is shifted downwards, and the bicycle model is shifted upwards). This exact offset again depends on the position in the image, and is extracted simultaneously with the generation of the model map. They are visualised in figure 2 as the *Offset maps*. To emphasize the center location the map is weighted with $G(\mathbf{x})$, centered at the image patch:

$$G(\mathbf{x}) = \alpha \left[e^{\left(-\frac{x^2}{2\sigma^2}\right)} - 1 \right] \qquad (2)$$

Where α indicates the penalty at the image borders (we empirically determined $\alpha = 2$).

3.4. Map exploration, NMS and tracking

Finally, these ROI probability maps are integrated in a *tracking-by-detection* framework to improve the accuracy and detection speed. This is done as follows. We define default search points (corresponding to search ROIs) at strategic positions in the image w.r.t. the blind spot zone. These positions are evaluated every frame using our pipeline mentioned above. Each probability map is then thresholded to extract local maxima. Next we perform *non-maxima suppression* (NMS) to cope with overlapping detections - using a variant of the 50% intersection criterion [9] - keeping only the best scoring detections. For each new detection a Kalman filter is instantiated. We employ a constant velocity motion model with state vector $x_k = \begin{bmatrix} x & y & v_x & v_y \end{bmatrix}^T$ containing the center of mass of each detection and the velocity. For consecutive frames, we predict the future location of the tracked instances, and use these predicted ROI centers (together with the default search points) as input to our detection pipeline. For each tracked instance we verify if a new detection is found in a circular region around the estimated location of which the radius is based on the scale at that location. We match the closest detection based on the Euclidean distance. If a match is found, the Kalman filter is updated and the new position is predicted. If no match is found, the Kalman tracker is updated based on the estimated position. If no detection is associated for multiple frames in a row, this track is discarded. Evidently, for new detections without existing tracker, a new track is instantiated. A qualitative tracking result is shown in fig. 4.

4. Experiments and Results

We performed extensive experiments to validate both the accuracy and speed of our algorithm. For this, we recorded several simulated dangerous blind spot scenarios with a genuine blind-spot camera mounted on a real truck, involving both pedestrians and bicyclists. A commercial blind spot camera was used (Orlaco115°), which has a viewing angle of 115 degrees and outputs images with a resolution of 640×480 at 15 frames per second. In total seven different scenarios were recorded each in which the truck driver makes a right turn, and the vulnerable road users act differently (*e.g.* the truck driver notices the VRUs and lets them pass, or the truck driver keeps driving simulating a near-accident). Our total testset consists of about 5000 frames, in which over 3600 pedestrians and 2400 bicyclists were manually labelled. Our framework is mainly implemented in Matlab with time-consuming parts (*e.g.* the detection and homography) in both C and OpenCV, and the hardware consists of an Intel Xeon E5 CPU at 3.1 GHz. As default search regions we define two entry points at the left, one entry point at the end of the truck, and one point in the blind spot zone (*i.e.* to recover lost tracks). These default search regions are indicated with a black star (*) in the left frame in fig. 2. Figure 5 displays the accuracy of our algorithm using a *precision-recall* curve (black curve). We achieve excellent accuracy (Average Precision of 81.2%). We also com-

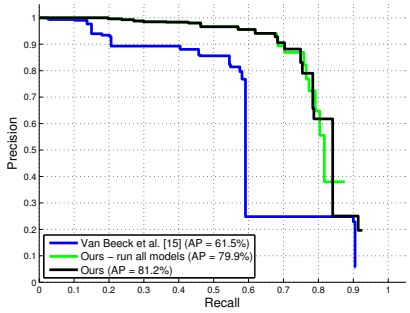

Figure 5. Accuracy results of our algorithm.

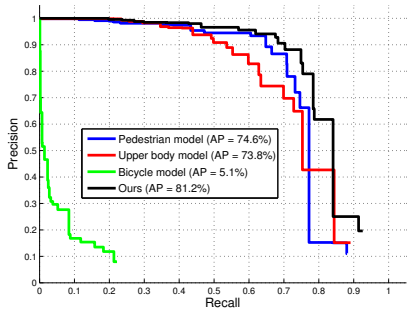

Figure 6. Accuracy improvement with multiple detection models.

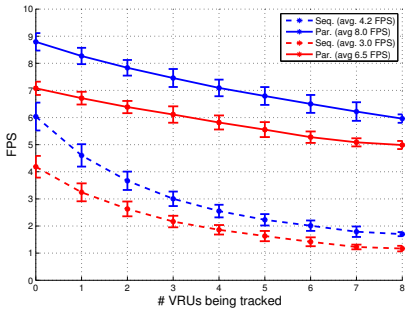

Figure 7. Processing speeds of our algorithm.

pare our algorithm with a vanilla implementation of [15] (blue curve). There we presented a similar safety system, however only targeting pedestrians. As seen, our algorithm easily outperforms this work on these datasets that also contain bicyclists. The green curve plots the accuracy when the model selection is discarded, and thus all detection models are evaluated at each location. The accuracy difference is minimal, indicating that our model selection procedure using the Model Map of fig. 3 is optimal. However, running all models evidently increases computation time. Next we validated the accuracy when running only individual models, shown in fig. 6. As observed, the accuracy increases when combining all three detection models. We performed several computational speed experiments. Although we mainly focused on high accuracy, during algorithmic development we aimed at keeping computational complexity (within the limitations of Matlab) as minimal as possible. Figure 7 displays the execution speed in function of the number of tracked VRUs in a frame. Evidently, the computation time increases with multiple tracks. To partially cope with this, we implemented both a sequential (dotted lines) and a parallel version (solid lines) of our framework. Parallel processing is achieved by processing each search region in a separate thread. The blue curves indicate our implementation with model selection, the red curves indicate processing speeds when evaluating all models. Our best implementation achieves an average processing speed of 8.0 fps.

5. Conclusions

We presented a multiclass object tracking framework targeting a specific application: detection and tracking pedestrians and bicyclists in the blind spot camera of a truck. This is a challenging task due to the non-standard viewpoint, high lens distortion, multiclass nature of this problem and the high accuracy demands. We propose the use of a warping window approach integrated with an efficient multiclass object detection scheme where we only run specific viewpoint detectors based on the position in the image. We achieve excellent accuracy results, while keeping the computational complexity adequate for practical applications.

References

[1] R. Benenson et al. Pedestrian detection at 100 frames per second. In *Proc. of CVPR*, 2012. 2

[2] R. Benenson, M. Omran, J. Hosang, and B. Schiele. Ten years of pedestrian detection, what have we learned? In *ECCV, CVRSUAD workshop*, 2014. 2, 3

[3] H. Cho et al. Real-time pedestrian detection with deformable part models. In *IEEE IV*, 2012. 2

[4] H. Cho, P. Rybski, and W. Zhang. Vision-based bicycle detection and tracking using a deformable part model and an EKF algorithm. In *Proc. of ITCS*, 2010. 2

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. of CVPR*, 2005. 2

[6] P. Dollár, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. *IEEE PAMI*, 36(8), 2014. 2

[7] P. Dollár, S. Belongie, and P. Perona. The fastest pedestrian detector in the west. In *Proc. of BMVC*, 2010. 2

[8] P. Dollár, Z. Tu, P. Perona, and S. Belongie. Integral channel features. In *Proc. of BMVC*, pages 91.1–91.11, 2009. 2

[9] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *In IEEE PAMI*, 34, 2012. 2, 4

[10] P. Felzenszwalb, R. Girschick, and D. McAllester. Cascade object detection with deformable part models. In *Proc. of CVPR*, pages 2241–2248, 2010. 3

[11] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Proc. of CVPR*, 2008. 2, 3

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CoRR*, 2013. 2

[13] I. Knight. A study of the implementation of dir. 2007/38/EC on the retrofitting of blind spot mirrors to HGVs, 2011. 1

[14] O. Russakovsky et al. Imagenet large scale visual recognition challenge. *In CoRR*, 2014. 2

[15] K. Van Beeck and T. Goedemé. Real-time pedestrian detection in a truck's blind spot camera. In *Proc. of ICPRAM*, 2014. 2, 3, 5

[16] J. Yan, X. Zhang, Z. Lei, S. Liao, and S. Z. Li. Robust multi-resolution pedestrian detection in traffic scenes. In *Proc. of CVPR*, pages 3033–3040. IEEE, 2013. 2